WHAT IS CLAIMED IS:

1. A method for correlating an encoded data word (X_0-X_{M-1}) with encoding coefficients (C_0-C_{M-1}), wherein each of (X_0-X_{M-1}) is represented by one or more bits and each said coefficient is represented by one or more bits, wherein each said coefficient has k possible states, wherein M is greater than 1, comprising the steps of:

   (1) multiplying X_0 with each state ($C_{0(0)}$ through $C_{0(k-1)}$) of said coefficient C_0, thereby generating results $X_0C_{0(0)}$ through $X_0C_{0(k-1)}$;

   (2) repeating step (1) for data bits (X_1-X_{M-1}) and corresponding said coefficients (C_1-C_{M-1}), respectively;

   (3) grouping said results of steps (1) and (2) into N groups and summing combinations within each of said N groups, thereby generating a first layer of correlation results;

   (4) grouping the results of step (3) and summing combinations of results within each group to generate one or more additional layers of results, and repeating this process until a final layer of results includes a separate correlation output for each possible state of the complete set of coefficients (C_0-C_{M-1}); and

   (5) comparing magnitudes output of said separate correlation outputs, thereby identifying a most likely code encoded on said data word.

2. The method according to claim 1, wherein steps (3) and (4) comprise the step of omitting summations that would result in invalid combinations of the encoding coefficients (C_0-C_{M-1}).

3. The method according to claim 1, further comprising the step of performing steps (1) through (5) using substantially the same hardware

for in-phase and quadrature phase components of the data word (X_0-X_{M-1}).

4. The method according to claim 1, wherein said coefficients (C_0-C_{M-1}) represent are real numbers.

5. The method according to claim 1, wherein said coefficients (C_0-C_{M-1}) represent complex numbers.

6. The method according to claim 1, wherein each said coefficients (C_0-C_{M-1}) is represented by a single bit.

7. The method according to claim 1, wherein each said coefficients (C_0-C_{M-1}) is represented by multiple bits.

8. The method according to claim 1, wherein said code patterns (C_0-C_{M-1}) represent a cyclic code keying ("CCK") code set substantially in accordance with IEEE 802.11 WLAN standard.

9. The method according to claim 8, wherein:

M equals 8;

each said coefficient (C_0-C_{M-1}) has two states, plus and minus;

N equals 4;

said first level of results comprises at least a portion of the following;

$(X_0C_0 + X_1C_1)$, $(X_0(-C_0)+X_1C_1)$, $(X_0C_0+X_1(-C_1))$, $(X_0(-C_0) + X_1(-C_1))$, $(X_2C_2 + X_3C_3)$, $(X_2(-C_2)+X_3C_3)$, $(X_2C_2+X_3(-C_3))$, $(X_2(-C_2)+(X_3(-C_3))$, $(X_4C_4 + X_5C_5)$, $(X_4(-C_4)+X_5C_5)$, $(X_4C_4+X_5(-C_5))$, $(X_4(-C_4) + X_5(-C_5))$, $(X_6C_6 + X_7C_7)$, $(X_6(-C_6)+X_7C_7)$, $(X_6C_6+X_7(-C_7))$, and $(X_6(-C_6)+(X_7(-C_7))$; and

wherein said second level of results comprises at least a portion of the following:

$((X_0C_0 + X_1C_1) + (X_2C_2 + X_3C_3))$, (i.e., B_0),
$((X_0C_0 + X_1C_1) + (X_2(-C_2)+X_3C_3))$, (i.e., B_1),
$((X_0C_0 + X_1C_1) + (X_2C_2+X_3(-C_3))$, (i.e., B_2),
$((X_0C_0 + X_1C_1) + (X_2(-C_2)+(X_3(-C_3))$, (i.e., B_3),

$((X_0(-C_0)+X_1C_1)+ (X_2C_2 + X_3C_3))$, (i.e., B_4)
$((X_0(-C_0)+X_1C_1) + (X_2(-C_2)+X_3C_3))$, (i.e., B_5),
$((X_0(-C_0)+X_1C_1) + (X_2C_2+X_3(-C_3))$, (i.e., B_6),
$((X_0(-C_0)+X_1C_1) + (X_2(-C_2)+(X_3(-C_3))$, (i.e., B_7),

$((X_0C_0+X_1(-C_1)) + (X_2C_2 + X_3C_3))$, (i.e., B_8)
$((X_0C_0+X_1(-C_1)) + (X_2(-C_2)+X_3C_3))$, (i.e., B_9),
$((X_0C_0+X_1(-C_1)) + (X_2C_2+X_3(-C_3))$, (i.e., B_{10}),
$((X_0C_0+X_1(-C_1)) + (X_2(-C_2)+(X_3(-C_3))$, (i.e., B_{11}),

$((X_0(-C_0) + X_1(-C_1)) + (X_2C_2 + X_3C_3))$, (i.e., B_{12})
$((X_0(-C_0) + X_1(-C_1)) + (X_2(-C_2)+X_3C_3))$, (i.e., B_{13}),
$((X_0(-C_0) + X_1(-C_1)) + (X_2C_2+X_3(-C_3))$, (i.e., B_{14}),
$((X_0(-C_0) + X_1(-C_1)) + (X_2(-C_2)+(X_3(-C_3))$, (i.e., B_{15}),

$((X_4C_4 + X_5C_5) + (X_6C_6 + X_7C_7))$, (i.e., B_{16}),
$((X_4C_4 + X_5C_5) + (X_6(-C_6)+X_7C_7))$, (i.e., B_{20}),
$((X_4C_4 + X_5C_5) + (X_6C_6+X_7(-C_7))$, (i.e., B_{24}),
$((X_4C_4 + X_5C_5) + (X_6(-C_6)+(X_7(-C_7))$, (i.e., B_{28}),

$((X_4(-C_4)+X_5C_5)+ (X_6C_6 + X_7C_7))$, (i.e., B_{17})
$((X_4(-C_4)+X_5C_5) + (X_6(-C_6)+X_7C_7))$, (i.e., B_{21}),
$((X_4(-C_4)+X_5C_5) + (X_6C_6+X_7(-C_7))$, (i.e., B_{25}),
$((X_4(-C_4)+X_5C_5) + (X_6(-C_6)+(X_7(-C_7))$, (i.e., B_{29}),

$((X_4C_4+X_5(-C_5)) + (X_6C_6 + X_7C_7))$, (i.e., B_{18})

$((X_4C_4+X_5(-C_5)) + (X_6(-C_6)+X_7C_7))$, (i.e., B_{22}),
$((X_4C_4+X_5(-C_5)) + (X_6C_6+X_7(-C_7))$, (i.e., B_{26}),
$((X_4C_4+X_5(-C_5)) + (X_6(-C_6)+(X_7(-C_7))$, (i.e., B_{30}),

$((X_4(-C_4) + X_5(-C_5)) + (X_6C_6 + X_7C_7))$, (i.e., $_{19}$)
$((X_4(-C_4) + X_5(-C_5)) + (X_6(-C_6)+X_7C_7))$, (i.e., B_{23}),
$((X_4(-C_4) + X_5(-C_5)) + (X_6C_6+X_7(-C_7))$, (i.e., B_{27}),
$((X_4(-C_4) + X_5(-C_5)) + (X_6(-C_6)+(X_7(-C_7))$, (i.e., B_{31}).

10. The method according to claim 9, wherein said second level of results omits one or more of B_0 through B_{31}.

11. The method according to claim 9, wherein said second level of results omits one or more of B_0 through B_{31} that represent invalid combinations of one or more of $(C_0\text{-}C_{M-1})$.

12. The method according to claim 9, wherein said second level of results omits one or more of B_0 through B_{31} where the omitted combination(s) would be redundant based on said CCK code specification.

13. The method according to claim 9, wherein said second level of results omits B_{24} through B_{31}.

14. The method according to claim 13, wherein said final level of results comprises:

$(B_0 + B_{19})$, $(B_0 + B_{21})$, $(B_1 + B_{20})$, $(B_1 + B_{18})$, $(B_1 + B_{23})$, $(B_2 + B_{20})$, $(B_2 + B_{17})$, $(B_2 + B_{23})$, $(B_3 + B_{16})$, $(B_3 + B_{22})$, $(B_4 + B_{17})$, $(B_4 + B_{18})$, $(B_4 + B_{23})$, $(B_5 + B_{16})$, $(B_5 + B_{22})$, $(B_6 + B_{21})$, $(B_6 + B_{19})$, $(B_7 + B_{20})$, $(B_7 + B_{17})$, $(B_7 + B_{18})$, $(B_8 + B_{20})$, $(B_8 + B_{17})$, $(B_8 + B_{18})$, $(B_9 + B_{21})$, $(B_9 + B_{19})$, $(B_{10} + B_{16})$, $(B_{10} + B_{22})$, $(B_{11} + B_{17})$, $(B_{11} + B_{18})$, $(B_{11} + B_{23})$, $(B_{12} + B_{16})$, $(B_{12} + B_{22})$, $(B_{13} + B_{20})$,

$(B_{13} + B_{17})$, $(B_{13} + B_{23})$, $(B_{14} + B_{20})$, $(B_{14} + B_{18})$, $(B_{14} + B_{23})$, $(B_{15} + B_{21})$, and $(B_{15} + B_{19})$.

15. The method according to claim 13, wherein said final level of results consists of:

$(B_0 + B_{19})$, $(B_0 + B_{21})$, $(B_1 + B_{20})$, $(B_1 + B_{18})$, $(B_1 + B_{23})$, $(B_2 + B_{20})$, $(B_2 + B_{17})$, $(B_2 + B_{23})$, $(B_3 + B_{16})$, $(B_3 + B_{22})$, $(B_4 + B_{17})$, $(B_4 + B_{18})$, $(B_4 + B_{23})$, $(B_5 + B_{16})$, $(B_5 + B_{22})$, $(B_6 + B_{21})$, $(B_6 + B_{19})$, $(B_7 + B_{20})$, $(B_7 + B_{17})$, $(B_7 + B_{18})$, $(B_8 + B_{20})$, $(B_8 + B_{17})$, $(B_8 + B_{18})$, $(B_9 + B_{21})$, $(B_9 + B_{19})$, $(B_{10} + B_{16})$, $(B_{10} + B_{22})$, $(B_{11} + B_{17})$, $(B_{11} + B_{18})$, $(B_{11} + B_{23})$, $(B_{12} + B_{16})$, $(B_{12} + B_{22})$, $(B_{13} + B_{20})$, $(B_{13} + B_{17})$, $(B_{13} + B_{23})$, $(B_{14} + B_{20})$, $(B_{14} + B_{18})$, $(B_{14} + B_{23})$, $(B_{15} + B_{21})$, and $(B_{15} + B_{19})$.

16. The method according to claim 9, wherein said final level of results omits one or more possible combinations of B_0 through B_{31}.

17. The method according to claim 9, wherein said final level of results omits one or more combinations of B_0 through B_{31} that represent invalid combinations of one or more of $(C_0\text{-}C_{M-1})$.

18. The method according to claim 9, wherein said final level of results omits one or more combinations of B_0 through B_{31} where the omitted combination(s) would be redundant based on a code specification.

19. The method according to claim 9, wherein said final level of results omits one or more combinations B_{24} through B_{31} where the omitted combination(s) would be invalid based on a code specification.

20. The method according to claim 1, further comprising the step of:

(7) performing an equalization process during one or more of steps (3) and (4).

21. The method according to claim 1, further comprising the step of:

    (7) performing an MLSE process during one or more of steps (3) and (4).

22. The method according to claim 1, further comprising the step of:

    (7) performing an adaptive process during one or more of steps (3) and (4).

23. The method according to claim 1, further comprising the step of:

    (7) performing an adaptive equalization process during one or more of steps (3) and (4).

24. The method according to claim 1, wherein one or more of (C_0-C_{M-1}) are constants.

25. The method according to claim 1, wherein one or more of (C_0-C_{M-1}) are variable.

26. The method according to claim 1, wherein steps (3) and (4) are implemented in accordance with:

$$N = \frac{n!}{r!(n-r)!} - L$$

wherein:

n represents a number of summer inputs;

r represents a number of summing inputs per kernal; and

L represents a number of invalid combinations

27. A system for correlating an encoded data word (X_0-X_{M-1}) with encoding coefficients (C_0-C_{M-1}), wherein each of (X_0-X_{M-1}) is represented by one or more bits and each said coefficient is represented by one or more bits, wherein each said coefficient has k possible states, wherein M is greater than 1, comprising:

inputs for each of (X_0-X_{M-1});

a multiplier coupled to each said input;

N summers, each coupled to a different group of outputs of said multipliers, whereby outputs of said N summers form a first layer of correlation results;

one or more additional layers of summers, each said additional layer of summers coupled to outputs of a previous layer of correlation results, said one or more additional layers of summers including a final layer of summers having a final layer of results including a separate correlation output for each possible state of the complete set of coefficients (C_0-C_{M-1}); and

a magnitude comparator coupled to said final layer of results.

28. A system for correlating an encoded data word (X_0-X_{M-1}) with encoding coefficients (C_0-C_{M-1}), wherein each of (X_0-X_{M-1}) is represented by one or more bits and each said coefficient is represented by one or more bits, wherein each said coefficient has k possible states, wherein M is greater than 1, comprising:

means for multiplying X_0 with each state ($C_{0(0)}$ through $C_{0(k-1)}$) of said coefficient C_0, thereby generating results $X_0C_{0(0)}$ through $X_0C_{0(k-1)}$;

means for repeating step (1) for data bits (X_1-X_{M-1}) and corresponding said coefficients (C_1-C_{M-1}), respectively;

means for grouping said results of steps (1) and (2) into N groups and summing combinations within each of said N groups, thereby generating a first layer of correlation results;

means for grouping the results of step (3) and summing combinations of results within each group to generate one or more additional layers of

results, and repeating this process until a final layer of results includes a separate correlation output for each possible state of the complete set of coefficients (C_0-C_{M-1}); and

means for comparing magnitudes output of said separate correlation outputs, thereby identifying a most likely code encoded on said data word.

29. A method for parallel correlation detection, comprising the steps of:

(1) receiving noisy input samples X_0, X_1, X_2, X_3, X_4, X_5, X_6, and X_7 from which a code must be extracted;

(2) forming four sets of sample pairs (X_0, X_1), (X_2, X_3) , (X_4, X_5), and (X_6, X_7) from said input samples;

(3) forming four correlation kernels ($X_iC_i + X_jC_j$), ($-X_iC_i + X_jC_j$), ($X_iC_i - X_jC_j$), and ($-X_iC_i - X_jC_j$) for each set of sample pairs formed in step (2), wherein X_i and X_j represent one of the four sample pairs formed in step (2) and wherein C_i and C_j represent predetermined weighting factors;

(4) combining the correlation kernels formed in step (3) to form a fast correlation transform trellis with sixty-four eight-tuple options; and

(5) using the sixty-four eight-tuple options formed in step (4) to extract the code from the input samples received in step (1).

30. A system for parallel correlation detection, comprising:

a module for receiving noisy input samples X_0, X_1, X_2, X_3, X_4, X_5, X_6, and X_7 from which a code must be extracted;

a module for forming four sets of sample pairs (X_0, X_1), (X_2, X_3) , (X_4, X_5), and (X_6, X_7) from said input samples;

a module for forming four correlation kernels $(X_iC_i + X_jC_j)$, $(-X_iC_i + X_jC_j)$, $(X_iC_i - X_jC_j)$, and $(-X_iC_i - X_jC_j)$ for each set of sample pairs formed in step (2), wherein X_i and X_j represent one of the four sample pairs formed in step (2) and wherein C_i and C_j represent predetermined weighting factors; and

a module for combining the correlation kernels formed in step (3) to form a fast correlation transform trellis with sixty-four eight-tuple options.

31. A method for correlating an encoded data word $(X_0\text{-}X_{M-1})$ with encoding coefficients $(C_0\text{-}C_{M-1})$, wherein each of $(X_0\text{-}X_{M-1})$ is represented by one or more bits and each said coefficient is represented by one or more bits, wherein each said coefficient has k possible states, wherein M is greater than 1, comprising the steps of:

(1) multiplying X_0 with states of said coefficient C_0;

(2) repeating step (1) for data bits $(X_1\text{-}X_{M-1})$ and corresponding said coefficients, respectively;

(3) grouping said results of steps (1) and (2) into N groups and summing combinations within each of said N groups, thereby generating a first layer of correlation results;

(4) grouping the results of step (3) and summing combinations of results within each group to generate one or more additional layers of results, and repeating this process until a final layer of results includes a correlation output for each possible state of the set of coefficients; and

(5) comparing magnitudes output of said correlation outputs, thereby identifying a most likely code encoded on said data word.

32. A method for parallel correlation detection, comprising the steps of:

(1) receiving noisy input samples from which a code must be extracted;

(2) forming at least four sets of sample pairs from said input samples;

(3) forming at least four correlation kernels for each set of sample pairs formed in step (2), wherein X_i and X_j represent one of the sample pairs formed in step (2) and wherein C_i and C_j represent predetermined weighting factors;

(4) combining the correlation kernels formed in step (3) to form a fast correlation transform trellis with at least sixty-four eight-tuple options; and

(5) using the at least sixty-four eight-tuple options formed in step (4) to extract the code from the input samples received in step (1).